Closed Form Expressions for the Performance Metrics of Data Services in Cellular Networks

Paolo Castagno,* Vincenzo Mancuso,† Matteo Sereno,* Marco Ajmone Marsan,‡†
*Università di Torino, Turin, Italy
†IMDEA Networks Institute, Madrid, Spain
‡Politecnico di Torino, Turin, Italy

Abstract

In this paper we study the queuing system that describes the operations of data services in cellular networks, e.g., UMTS, LTE/LTE-A, and most likely the forthcoming 5G standard. The main characteristic of all these systems is that after service access, resources remain allocated to the end user for some time before release, so that if the same user requests access to service again, before a system timeout, the same resources are still available. For the resulting queuing model, we express the blocking probability in closed form, and we also provide recursive expressions in the number of connections that can be handled by the base station. Closed form expressions are also derived for other useful performance metrics, i.e., throughput and network service time. Analytical results are validated against results of a detailed simulation model, and compared to traditional queueing models results, such as the Erlang B formula iteratively applied to the resources that are not blocked by potentially returning users. Our analysis complements the performance evaluation of the other key mechanism used to access data services in cellular networks, namely the random access, which precedes the resource allocation and utilization phase studied in this paper.

I. Introduction

Mobile data traffic is growing incredibly fast, experiencing a 18-fold growth between 2011 and 2016, and an impressive $+63\%$ in the last year only, as reported by the most recent *Global Mobile Data Traffic Forecast Update* by Cisco® [1]. The same report also shows how cellular speed increases even faster, with a $+340\%$ in 2016, resulting in average downlink speeds of 6.8 Mb/s. However, what is most remarkable is the number of connected mobile devices: 429 million mobile devices and connections were added in 2016, reaching a cumulative 8 billion figure.

In this context, the success of network operators depends on the performance of key critical services like the ones offered by dedicated apps for social media, banking/financial transactions, home and factory automation, health monitoring, etc., which are being increasingly ported to smartphones [2], [3]. For such services, as well as for the emerging Internet of Things (IoT) scenario [4], throughput is not necessarily the key metric, while outage (or "blocking probability") and access delay become fundamental [5], as it is commonly observed during crowded events or when an emergency situation arises [6]. Indeed, those performance figures are affected by the number of devices requesting network access in a cell [7], [8]. Thus, such number risks to become the new network bottleneck, as we discuss in this paper.

Access to resources in cellular network is basically the result of a two-step operation, which consists in first notifying the network about the user intention to join the network over a random access channel, and second, if the random access request was acknowledged by the base station, in negotiating a data channel [9]. Existing models focus on achievable throughput, and limit the analysis of resource access protocols to the random access operation in legacy cellular networks [7], [10] or in newer cells which support enhanced protocols for machine-type communications or IoT [8], [11]. By so doing, they neglect the blocking probability for the data channel negotiation, which is non-negligible when the number of data channels requested by users becomes comparable with the limited number of connections that a base station can handle [9]. A few other models follow the details of signaling channel operation, but do not work in heavily utilized cells, in which the finite amount of resources of the network becomes the bottleneck [12].

In this paper, we show that the data channel negotiation and utilization can be modeled independently of the random access, and therefore our analysis complements and completes existing studies on the performance of cellular networks. Specifically, we derive a chain of models for the system under study. We show that a queuing network with non-Bernoulli routing is needed to characterize the system in detail, which is not practical and does not lead to convenient analytical expressions. However, with a few approximations validated through simulation, we show that the system can also be modeled with a closed queueing network that admits a product-form solution which is independent of the distribution of service times. From this product-form queuing network we are able to derive closed form expressions for the blocking probability, the throughput and the network service time with multiple users competing for data transmission resources. In addition, we derive convenient recursive expressions for all proposed metrics, which are extremely useful to implement the formulas numerically, and avoid multiplications and divisions with extremely high or extremely low factors. We validate our model against simulation and compare the results with simpler approximations based on well-known approaches such as the Erlang-B formula. Our results

This work is partially supported by the European Commission through the H2020 5G-TRANSFORMER project (Project ID 761536) and by the Ramon y Cajal grant (ref: RYC-2014-16285) from the Spanish Ministry of Economy and Competitiveness.

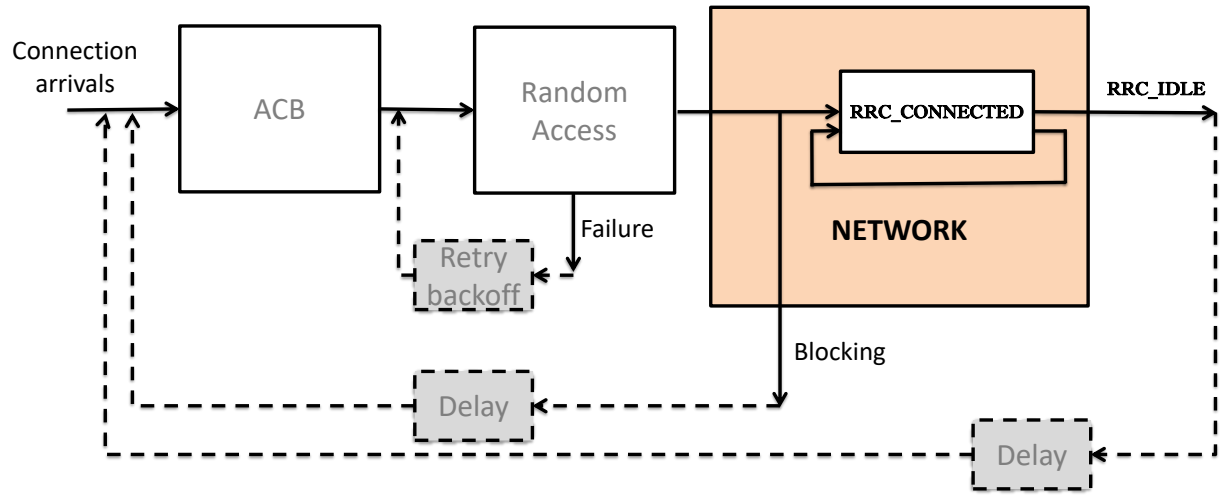

Fig. 1. High-level system view. Dotted lines and grey-shaded delay/backoff blocks indicate optional system components. Such optional components, jointly with ACB and Random Access, have been widely addressed by the scientific literature and are out of the scope of this work. The response of the NETWORK block to the load offered by the Random Access is analyzed in this paper, and characterized in terms blocking probability and service offered to the end users to which the base station allots cellular resources.

show that our closed form expressions are very accurate and outperform alternative methods in terms of robustness, complexity and precision.

II. System

Access to service in a cellular network like LTE/LTE-A is based on a random access procedure followed by a signaling exchange needed to synchronize base station and mobile device and assign resources to a connection over which data can be exchanged. The latter is the so-called RRC (Radio Resource Control) connect phase.

This is a general scheme adopted by 3GPP since the early days of cellular data networks, which will be also included in future LTE and 5G releases [9]. Fig. 1 offers a high-level view of the cellular system from the point of view of access to transmission resources. The system is characterized by a few blocks that can be analyzed independently. The activity of customers creates a feedback loop between some of such blocks (represented with dashed lines and shaded blocks in the figure), so that a detailed analysis of the behaviour of the system cannot be generalized, since it depends on how connection requests are generated and on how customers return to ask for new transmission resources after a period of inactivity or after a failed attempt.

The random access phase resembles a multi-channel slotted Aloha system [13] with, optionally, access class barring (ACB [14]). ACB makes access requests a non-persistent process, meaning that a customer ready to request access can be forced to defer its request with some probability. A slot in such system is the interval of time between two random access opportunities (RAOs), regularly announced by the base station. Note that, since in LTE and similar networks, resources are split over time into units called subframes, a RAO includes one or more subframes.

Once the customer attempts random access and succeeds, the following RRC connect phase consists of two parts: access resolution and connection establishment. During access resolution, customers are in `RRC_IDLE` state, and have no resource allocated to transmit and receive data. The ones that have passed the random access phase are then requested by the base station to specify the parameters of their connection request, i.e., they are invited by the base station to proceed with the connection request. More specifically, 3GPP standards specify that in the random access phase, customers announce their willingness to access resources by sending an anonymous orthogonal code during the RAO, picked from a restricted dictionary. If two or more customers pick the same code, the base station decodes it only once and asks whoever has sent that code to specify its connection request parameters in a given set of time-frequency resources. At that point, if two or more customers have used the same code, their requests collide and have to wait for one of the next RAOs to attempt access again. From a logical point of view, the random access phase effectively concludes with the access resolution. This procedure is equivalent to excluding from access grant requests all customers that have used the same orthogonal code during the random access phase, so that random access and access resolution can be jointly analyzed as a regular multi-channel slotted Aloha system. The performance of such system has been widely studied in the literature, so in this paper we focus on the connection establishment part, which happens in the subsystem indicated as NETWORK in Fig.1, and which has been so far oversimplified or completely neglected in the evaluation of cellular performance.

The base station can only decode and acknowledge a limited number of requests per subframe and can only handle a limited number of simultaneous connections. So, only a fraction of non-collided requests involved in the connection establishment procedure receive resources and are promoted to the RRC_CONNECTED state [9]. No new connection request can be acknowledged when the limit of connections has been reached. 3GPP standards further specify that, once a connection is established, it remains active until a timeout expires after service completion. After that, it returns to the RRC_IDLE state. Therefore, recently served customers keep holding their resources for a short while (decided by the network operator, typically between 1 second and 1 minute) after completing their traffic exchange, which could prevent freshly arrived customers to obtain service even if the actual number of active connections is below the maximum.

In the operation of the system, customers that pass the random access phase are subject to a blocking probability because of the limitation in the number of RRC_CONNECTED customers. For what concerns the service rate received by customers, this is a complex function of the scheduler policy implemented at the base station, the quality of wireless links, and the number of active connections. For tractability reasons, here we assume that the base station adopts a processor sharing policy with a single class of users, so that resources are shared equally. Moreover, we assume that all customers can use the same modulation and coding scheme. Such assumptions are realistic for M2M communications and in small cell environments in general.

The system we have described is characterized by two kinds of arrivals: the *exogenous* flow generated by the random access system and the *endogenous* flow generated by customers that return before their RRC_CONNECTED timeout expires. Similarly, customers leave the system for two reasons: if blocking occurs over the exogenous arrival flow, and if, after receiving service, customers do not generate traffic before the expiration of the RRC_CONNECTED timeout, so that they fall back to the RRC_IDLE state.

From the above description, we see that it is important to track both the number of active connections as well as the number of established yet inactive connections, since those values determine blocking probability and service rate. Next, we will show how to model the system and derive closed form expressions for the blocking probability suffered by the exogenous arrival flow and for other key performance indicators such as the average service rate and the throughput of the cellular system.

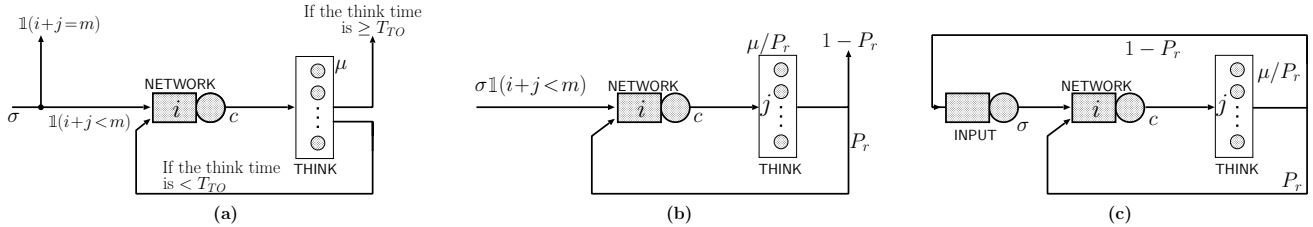

Fig. 2. Queuing network models for the connection establishment process in a cellular network (NETWORK subsystem in Fig.1): network with non-Bernoulli routing (a), open network with population constraint (b), closed network (c)

III. A Chain of Queueing Network Models

The system that was described in the previous section can be abstracted into a model where requests for service are submitted by end user terminals. Requests can be refused access to network resources if those are not readily available, and otherwise enter into a service phase. At the end of service, resources remain associated to the end user for a time T_{TO}, before being released. If during this time T_{TO} the end user issues another request, he/she is immediately granted the same resources he/she had before. Otherwise, resources are freed, and a new request by the same user will be accepted conditionally on the availability of free resources in the network.

We can model this behavior with the network of queues in Fig. 2(a). Here, a Poisson process with rate σ feeds the queue *NETWORK*, which has one exponential server operating with rate c according to the processor sharing discipline. The assumptions of Poisson arrivals and exponential service times will be relaxed later on. Both σ and c are expressed in services/s. At the end of service, customers enter the infinite-server queue *THINK*, with exponential service at rate μ, expressed in s^{-1} units.

When service at queue *THINK* ends, customers are routed back to queue *NETWORK* if the instance of their service time was shorter than T_{TO}, and they leave the network of queues otherwise (with a service time in *THINK* equal to T_{TO} or longer). At any given time instant, the number of customers at queue *NETWORK* is denoted by i, and the number of customers in queue *THINK* whose received service is shorter than T_{TO} is denoted by j (j thus is *not* the total number of customers at queue *THINK*). Arriving customers are lost when the sum of the numbers i and j is equal to m.

It is important to observe that in the network of queues we just described the state depends on accrued service times, and the customer routing is not governed by Bernoulli choices, but depends on service time instances; hence, this model cannot be solved by using the classical queueing network solution methods (e.g., Markovian analysis, or product-form solution).

We can approximate the behavior of the queuing network model we just described with another queueing network, as in Fig. 2(b). In this second model, the infinite-server queue *THINK* has exponential service at rate μ, truncated at T_{TO}. That is, the pdf of the service time at queue *THINK* is

$$f_{T_{\text{THINK}}}(t) = \mu e^{-\mu t}[u(t) - u(t - T_{TO})] + (1 - P_r)\delta(t - T_{TO}) \tag{1}$$

which results in an average service time

$$E[T_{\text{THINK}}] = P_r/\mu. \tag{2}$$

with

$$1 - P_r = e^{-\mu T_{TO}}, \tag{3}$$

where $u(\cdot)$ is the unit step function, and $\delta(\cdot)$ is the Dirac delta function.

In this new network of queues we introduce a Bernoulli choice for customer routing after the service at queue *THINK*, with a probability P_r of re-accessing the network resources, which is equal to the probability of the service at queue *THINK* ending before time T_{TO}.

Arriving customers are lost when the sum of the numbers of customers at queues *NETWORK* (denoted by i) and *THINK* (j) is equal to m.

This modification in the model implies an approximation, which calls for a validation against simulation results, as we will do in a later section of this paper.

Because of the fact that the total number of customers in the previous model is limited to m, we can transform our open queuing network into a closed queuing network, by adding one more single-server queue (named *INPUT*), with exponential service at rate σ and processor sharing discipline, so as to obtain the model in Fig. 2(c). Note that this third queuing network model, with a total customer population of size m, is exactly equivalent to the previous one, and that the exponentially distributed service time of *THINK*, truncated at T_{TO} is still there.

This last model allows the exploitation of a classical queueing network result concerning service time distributions. In particular, the well known BCMP product-form theorem [15] states that the queueing network model of Fig. 2(c) has product-form solution whenever the service times of single server queues with processor sharing discipline, or of infinite server queues (the former is the case of queues *INPUT* and *NETWORK*; the latter is the case of *THINK*), has a rational Laplace transform. Probability distributions with rational Laplace transform form a very wide class that includes matrix exponential, coxian, and phase-type distributions (see for instance [16]). This translates into the fact that *any* distribution that can be expressed in phases (e.g., phase-type) is acceptable as a service time distribution in this queuing network while preserving the product-form solution. Hence, by playing with phase-type distributions we can model a very wide class of arrival processes and service time distributions, in either an exact or an approximate manner. For instance, a deterministic service time cannot be exactly represented by using a phase-type distribution, but it can be well approximated by using an Erlang distribution with a large number of phases. A similar thing can be done for a exponential delay, truncated at T_{TO}. As we will show when we address computational issues, the number of phases does not affect the solution complexity, since the performance measures we are interested in only require the specification of the average service times. For the model in Fig. 2(c) we can therefore just specify an average delay P_r/μ for the *THINK* queue, and a processor sharing discipline for the *NETWORK* and *INPUT* queues, with no loss of generality. In this manner we can model a very wide class of arrival processes and service time distributions.

For this queuing network model, we can define the state as

$$s_{i,j,k} = (i, j, m - i - j),$$

where i is the number of customers in queue *NETWORK*, j is the number of customers in queue *THINK*, and $k = m - i - j$ is the number of customers in queue *INPUT*. We can then compute the equilibrium probabilities as the product of the state probabilities of the three queues computed for unit arrival rate at the *NETWORK* queue (note that any arrival rate can be used and that the arrival rate at the *THINK* queue is the same as at the *NETWORK* queue, while queue *INPUT* sees $(1 - P_r)$ times the arrival rate of the other queues):

$$\pi_{(i,j,m-i-j)} = \frac{1}{G}\left(\frac{1}{c}\right)^i \frac{1}{j!}\left(\frac{P_r}{\mu}\right)^j \left(\frac{1 - P_r}{\sigma}\right)^{m-i-j} \tag{4}$$

where G is a normalization constant. In the following, to emphasize the dependency of the normalization constant G with respect to parameter m we will use the notation $G(m)$.

A. Recursive Equations for Performance Metrics

Dealing with a queuing network that admits a product-form solution allows the use of several different computational algorithms developed for this class of models. In particular, we can apply the extremely effective algorithm known as Mean Value Analysis (MVA) [17], that allows the computation of performance measures through a set of recursive equations. In the

following, we provide the MVA equations for the queueing network depicted in Fig. 2(c) where m customers cyclically visit the three queues.

For convenience, we label the three queues as $I = 1$, $N = 2$ and $T = 3$, respectively for *INPUT*, *NETWORK* and *THINK*, and in the rest of the paper we interchangeably use letters and numbers as indexes, depending on the context (e.g., ξ_2 and ξ_N will both refer to the throughput of *NETWORK*). The parameters $S_i, i \in \{1, 2\}$, represent the inverse of the service rates for queues *INPUT* and *NETWORK*, whereas Z_3 represents the average delay for the infinite-server queue *THINK* (e.g., $S_1 = 1/\sigma$, $S_2 = 1/c$, and $Z_3 = P_r/\mu$).

We denote by $\boldsymbol{P}$ the customer routing matrix, where the element $p_{i,j}$ represents the probability that a customer completing service at queue i moves next to queue j for service. From $\boldsymbol{P}$ we can compute the visit ratio vector $\boldsymbol{v}$ as $\boldsymbol{v} = \boldsymbol{v}\boldsymbol{P}$. The element V_i of $\boldsymbol{v}$ represents the average relative number of visits of a customer to queue i, and does not depend on the population size.

The performance indexes which summarize the behavior of the queueing network at steady-state are, with population m:

- the queue *utilization* $U_i(m)$, i.e., the fraction of time in which the server of queue i is busy;
- the cumulative *mean sojourn time* $W_i(m)$ experienced by a customer while waiting and subsequently receiving service at queue i;
- the average number of customers (waiting and in service) at a queue, $Q_i(m)$;
- the system throughput $\xi(m)$, and the queue throughput $\xi_i(m)$ (with $\xi_i(m) = V_i\xi(m)$).

These quantities, for $i = 1, 2$, can be computed as follows:

$$W_i(m) = S_iV_i(1 + Q_i(m-1)), \tag{5}$$

$$\xi(m) = \frac{m}{Z_3V_3 + \sum_{i=1}^{2} S_iV_i(1 + Q_i(m-1))} \tag{6}$$

$$U_i(m) = S_iV_i\xi(m), \tag{7}$$

$$Q_i(m) = U_i(m)(1 + Q_i(m-1)), \tag{8}$$

The above MVA formulas define a recursive algorithm in terms of the queue lengths $Q_i(m-1)$. The recursion starts from $m = 0$. In particular, we start with $Q_i(0) = 0$, from this we can derive $W_i(1) = S_iV_i$, and then we can derive $\xi(1)$, $U_i(1)$, and $Q_i(1)$. In this manner we can derive the performance indexes for the queueing network with m customers in $O(m)$ steps.

IV. Performance Metrics

In this section we introduce the set of metrics that characterize the performance of the system under study, and for these metrics we provide a definition in terms of the queueing network model of Fig. 2(c).

A. Blocking probability

The blocking probability at the base station under investigation is reflected in the probability that the sum of customers at queues *NETWORK* and *THINK* is equal to m (or, equivalently, the probability that queue *INPUT* is empty, which occurs when $i + j = m$). Therefore, the blocking probability of exogenous connection requests (proceeding from the Random Access block of Fig. 1) is:

$$P_b(m) = \sum_{j=0}^{m} \pi_{(m-j,j,0)} = 1 - U_I(m). \tag{9}$$

Note that the sum in the equation above yields a closed form expression for the blocking probability. Indeed, using (4) and normalizing the sum of probabilities, it is easy to compute the normalization constant $G(m)$, and hence also derive the blocking probability in closed form, as follows:

$$P_b(m) = \frac{\sum_{j=0}^{m} \frac{1}{j!} \left(\frac{\sigma P_r}{\mu(1-Pr)}\right)^j \left(\frac{\sigma}{c(1-Pr)}\right)^{m-j}}{\sum_{j=0}^{m} \frac{1}{j!} \left(\frac{\sigma P_r}{\mu(1-Pr)}\right)^j \sum_{i=0}^{m-j} \left(\frac{\sigma}{c(1-Pr)}\right)^i}. \tag{10}$$

B. Throughput

The throughput of the base station corresponds to the throughput of queue *NETWORK* (denoted as $\xi_N(m)$), which is equal to the throughput of queue *THINK* (denoted as $\xi_T(m)$), while the throughput of queue *INPUT* is $\xi_I(m) = (1 - P_r)\,\xi_N(m)$. It is thus enough to find an expression for the throughput of the *NETWORK* queue, which can be easily obtained by considering that the entire flow entering the queue is served, so that $\sigma(1 - P_b(m)) + P_r\xi_N(m) = \xi_N(m)$. Therefore, the following result holds:

$$\xi_N(m) = \sigma\frac{1 - P_b(m)}{1 - P_r}. \tag{11}$$

C. Average service time

The average time required to serve a request can be mapped into the average time spent at queue *NETWORK*. By using Little's law, we can derive this average time as:

$$W_N(m) = Q_N(m)/\xi_N(m). \tag{12}$$

Note that, since the *NETWORK* queue uses a processor sharing policy with no waiting room, the average time spent in the queue can be (equivalently) derived as

$$W_N(m) = \frac{1}{c}\frac{\sum_{i=1}^{m} iP_i(m)}{1-P_0(m)}, \tag{13}$$

where $P_i(m)$ is the probability to have i customers under service (for $i = 0, \ldots, m$). Eq. (13) simply uses the fact that resources are equally split among active customers, subject to the fact that at least one user is under service. If only a user is under service, $\frac{1}{c}$ is the average service time.

V. Recursive Expressions and Approximations

A. Recursive computation of the blocking probability

Let $A_k(m)$ be the probability of having k customers in the *INPUT* queue when the population of the closed queueing network is m. When the *INPUT* queue is empty, a blocking occurs in the cellular network. From the *Arrival Theorem* [18], we know that

$$A_1(m) = \frac{\xi_I(m)}{\sigma} A_0(m-1), \tag{14}$$

By simple inspection of the closed queueing network, the following relations hold:

$$\xi_I(m) = \sigma(1 - P_b(m)); \tag{15}$$

$$A_0(m-1) = P_b(m-1); \tag{16}$$

$$\Rightarrow A_1(m) = (1 - P_b(m))P_b(m-1). \tag{17}$$

We now look for a relation between $A_1(m)$ and $A_0(m)$. Using the expressions for state probabilities, we obtain the following expression, for $0 \leq k \leq m$:

$$\begin{aligned} A_k(m) &= \sum_{j=0}^{m-k} \pi_{(m-j-k,j,k)} \\ &= \frac{1}{G(m)} \sum_{j=0}^{m-k} \frac{1}{j!} \left(\frac{\sigma P_r}{\mu(1-Pr)}\right)^j \left(\frac{\sigma}{c(1-Pr)}\right)^{m-j-k}. \end{aligned} \tag{18}$$

From the above expression, evaluated for $k \in \{0, 1\}$, and having identified that $A_0(m) = P_b(m)$, the following relation holds:

$$A_1(m) = \frac{c(1-Pr)}{\sigma}\left(P_b(m) - \frac{1}{G(m)}\frac{1}{m!}\left(\frac{\sigma P_r}{\mu(1-Pr)}\right)^m\right), \tag{19}$$

which, once plugged in (17), leads to the following recursive expression for the computation of P_b:

$$P_b(m) = \frac{\frac{\alpha(m)}{G(m)}\frac{c(1-P_r)}{\sigma} + P_b(m-1)}{\frac{c(1-P_r)}{\sigma} + P_b(m-1)}, \tag{20}$$

where $\alpha(m) = \frac{1}{m!}\left(\frac{\sigma P_r}{\mu(1-P_r)}\right)^m$ can be computed recursively as

$$\alpha(m) = \frac{\sigma P_r}{m\mu(1-Pr)}\alpha(m-1). \tag{21}$$

Note that $\frac{\alpha(k)}{G(k)} \in [0, 1]$ for all positive values of k, so that the recursive expression of P_b always yields a value between 0 and 1. Moreover, there exists an interesting recursive expression for the computation of $G(m)$ to be used in (20):

$$G(m) = G(m-1) + \sum_{j=0}^{m} \frac{1}{j!}\left(\frac{\sigma P_r}{\mu(1-P_r)}\right)^j \left(\frac{\sigma}{c(1-P_r)}\right)^{m-j}.$$

Therefore, $P_b(m)$ can be computed recursively starting with the following initialization: $P_b(0) = 1$, $\alpha(0) = 1$, $G(0) = 1$.

B. Recursive computation of the service time

The time to serve a request corresponds to the sojourn time in queue *NETWORK*. Eq. (13) holds for the *NETWORK* queue with at most m services in progess. However, note that the average number of services in progress, given that at least one customer is under service, is $\frac{\sum_{i=1}^{m} iP_i(m)}{1-P_0(m)}$. This is also given by 1 service in progress plus the average number of services in progress in a system with $m-1$ possible services, that is $1+\sum_{i=1}^{m-1} iP_i(m-1)$. This result holds because the *Arrival Theorem* holds for closed queueing networks with product-form solution [17]. Applying this equality repeatedly leads to the following result:

$$W_N(m) = \frac{1}{c}\sum_{i=0}^{m-1}\prod_{j=1}^{i}\left[1-P_0(m-j)\right]. \tag{22}$$

Eq. (22) offers the possibility to compute the average time spent at queue *NETWORK* recursively, by analysing systems with at most $m-1$ parallel services, which can be useful for numerical implementations.

C. Relation between blocking probability and other metrics

There is a simple relation between the probability that the *NETWORK* queue is inactive and the blocking probability, denoted here as $P_0(m)$ and $P_b(m)$, respectively. Therefore, computing $P_b(m)$ is enough to compute also the time spent in *NETWORK*. Specifically, the following result holds:

$$\xi_N(m) = [1-P_0(m)]c; \tag{23}$$

$$\xi_N(m) = \sigma[1-P_b(m)] + P_r\xi_N(m); \tag{24}$$

$$\begin{aligned}\Rightarrow P_b(m) &= 1-\frac{\xi_N(m)\,(1-P_r)}{\sigma}\\ &= 1-\frac{c}{\sigma}(1-P_r)[1-P_0(m)].\end{aligned} \tag{25}$$

Moreover, from (22), the service time with at most m parallel services can be computed from $P_0(k)$ obtained for $k=1,2,\ldots,m-1$, as follows:

$$W_N(m) = \frac{1}{c}\sum_{i=0}^{m-1}\beta(i,m); \tag{26}$$

with $\beta(0,m)=1$ and $\beta(i,m)=\beta(i-1,m)(1-P_0(m-i))$.

Thus, to characterize the system it is enough to compute the set $\{P_b(k)\}_{0\le k\le m}$

D. Approximations

Next, we present a few closed form and iterative approaches to the analysis of the system, based on some simplifications and on the use of well-known formulas, commonly used in cellular system performance analysis and design. The results of these approaches will be compared to those of our model and of simulation in the next section.

Erlang B in closed form. We start with an approximation using the Erlang B formula for an $M/M/k/0$ system with arrival rate σ and $k=\lfloor m(1-P_r)\rfloor$ servers. In this case, customers are served in parallel with speed c/m each, and the blocking probability can be expressed in closed form as follows:

$$P_b = \text{ErlangB}\left(\frac{\sigma}{c/m}, \lfloor m(1-P_r)\rfloor\right). \tag{27}$$

By using this formula we assume that about mP_r servers are always busy due to returning customers, and service to exogenous requests can be provided by the remaining $m(1-P_r)$ servers with fixed rate (taken to be the same as the lowest individual service rate that can be experienced at queue *NETWORK*).

Iterative Erlang B. A second approximation based on an $M/M/k/0$ queue can be devised by choosing an arrival rate equal to $\sigma+P_r\xi_N$, and $k=m$ servers. Like before, each server runs at speed c/m. In this case, we account for the maximum number of services in progress, and for both exogenous and endogenous request flows.

This case requires iterations because the queue input and output are coupled by a feedback mechanism. Indeed, note that $\xi=\sigma\frac{1-P_b}{1-P_r}$, so that an iterative expression for the loss probability is as follows:

$$P_b = \text{ErlangB}\left(\frac{\sigma}{c/m}\frac{P_r}{1-P_r}(1-P_b), m\right). \tag{28}$$

Note also that the blocking probability is here approximated as the ratio between throughput and *all* arrivals, including returning customers, which in a real system cannot experience blocking.

$M/M/1/k$ **in closed form.** In this case, we approximate the system behavior as an M/M/1 queue with service rate equal to c, that can accept no more than k customers (in service or waiting). The limit on the number of customers is set as $k = \lfloor m(1 - P_r) \rfloor$. The blocking probability is:

$$P_b = \frac{1 - \frac{c}{\sigma}}{1 - \left(\frac{c}{\sigma}\right)^{\lfloor m(1-P_r)\rfloor+1}}. \tag{29}$$

Iterative $M/M/1/k$**.** This case is similar to the second one, with the loss probability formula of an $M/M/1/k$ system with processor speed equal to c and with $k = m$ servers instead of the Erlang B formula with the same number of serves and speed c/m. We use $\xi_N = \sigma\frac{1-P_b}{1-P_r}$ to obtain the following iterative expression:

$$P_b = \frac{1 - \frac{c}{\sigma\left(1+\frac{P_r}{1-P_r}(1-P_b)\right)}}{1 - \left(\frac{c}{\sigma\left(1+\frac{P_r}{1-P_r}(1-P_b)\right)}\right)^{m+1}}. \tag{30}$$

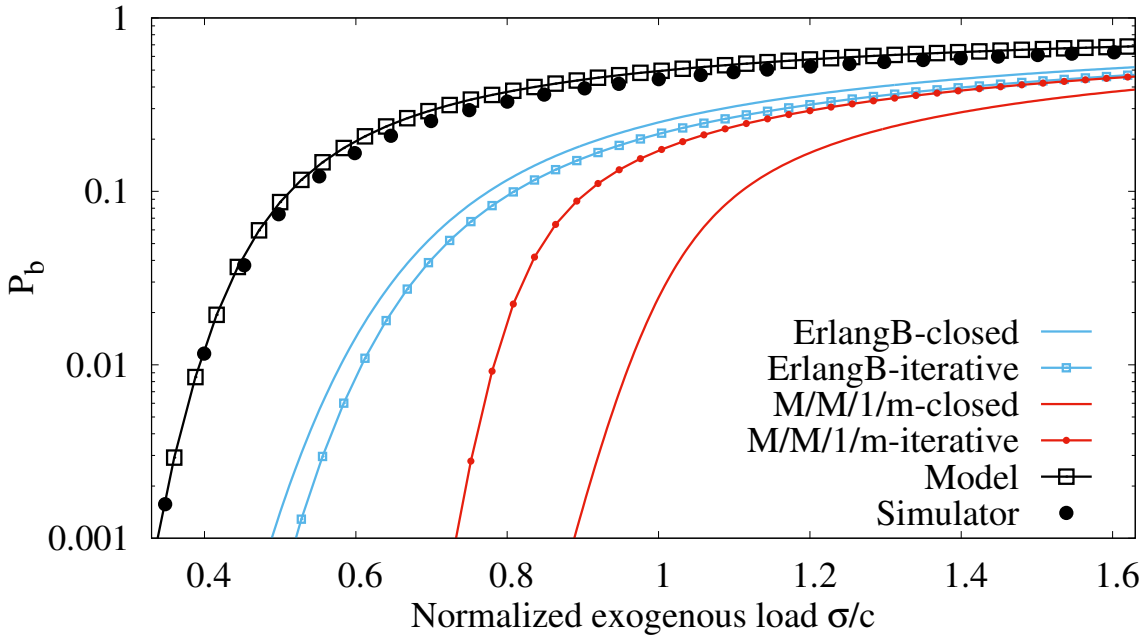

Fig. 3. Blocking probability computed with our model and with other alternative models, compared to simulation estimates ($m = 50$, $P_r = 0.2$). With small values of m, models that do not account for the presence of the *THINK* queue have low accuracy.

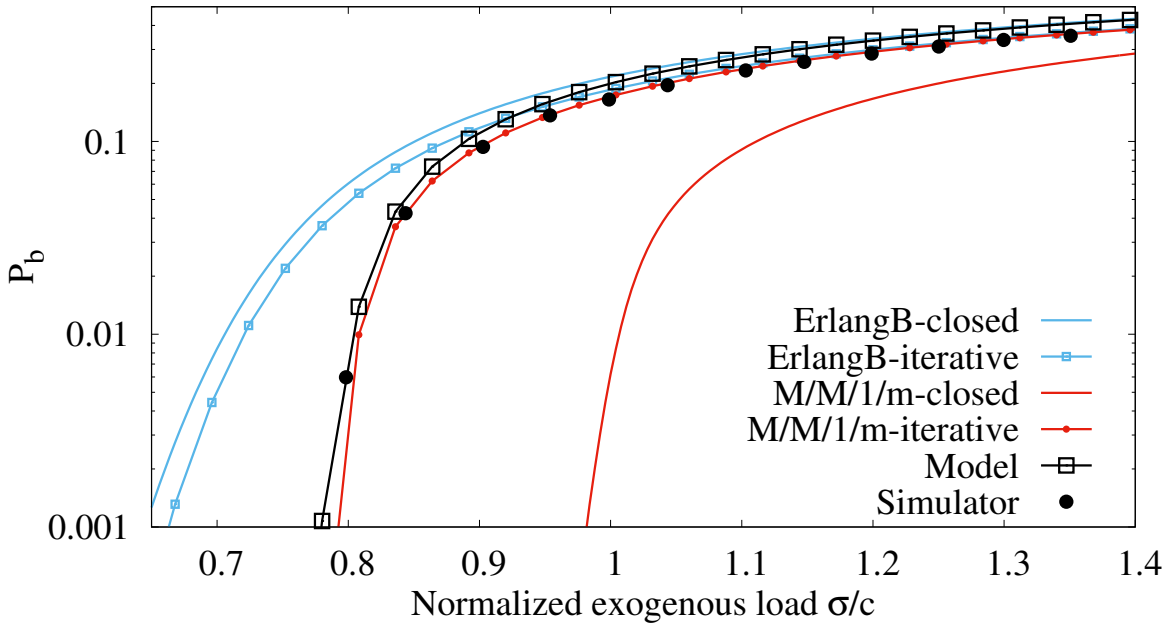

Fig. 4. Blocking probability computed with our model and with other alternative models, compared to simulation estimates ($m = 200$, $P_r = 0.2$). With high values of m, modeling the *THINK* queue becomes less important, and iterative methods become as good as our model.

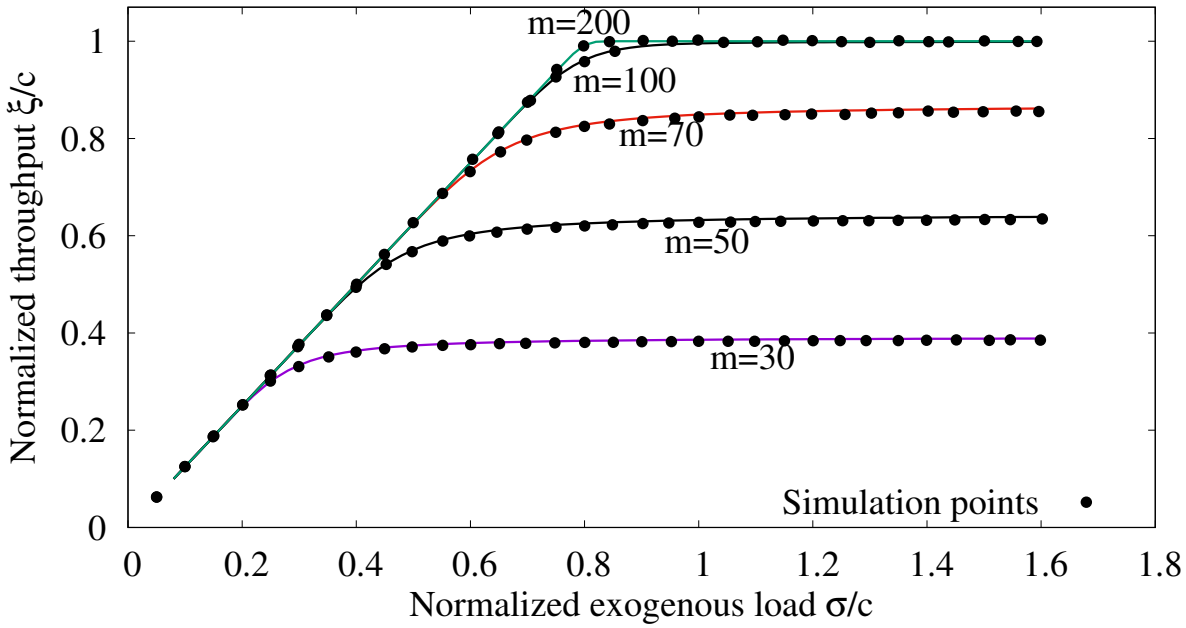

Fig. 5. System throughput (i.e., throughput of queue *NETWORK*), relative to the queue capacity c, for $\mu^{-1} = 30$ s and $P_r = 0.2$. The system throughput is monotonic in the exogenous arrival rate, and saturates to a value that depends on the maximum number of simultaneous services (the capacity of the queue is constant)

VI. Numerical results

We use a home-grown packet simulator written in python to validate our model and to assess the impact of our main approximations, namely: (i) the use of Bernoulli routing to make the queueing network of Fig. 2(a) tractable, and (ii) the replacement of load dependent service times in the NETWORK station with an exponential distribution with the same average (see Fig. 2).

From the expressions derived in Section IV, one can notice that blocking probability and throughput, as well as the time spent in *NETWORK*, depend on various parameters, namely the maximum number of services in progress m, the exogenous arrival rate σ, the service rate c, and the pair (μ, P_r) characterizing the *THINK* queue. Note that the parameters (μ, P_r) are equivalent to the pair $(1/\mu, T_{TO})$, that is the average time spent in *THINK* and the `RRC_CONNECTED` timeout. Moreover, results depend on the ratio σ/c rather than on their individual values. Such ratio represents the load offered by the exogenous arrivals only. In the following, we explore how such parameters affect system performance.

Validation and basic results on blocking and throughput. In the system under evaluation, blocking probability and throughput are not equivalent metrics, because of the presence of returning customers that cannot experience blocking. Therefore, we validate by simulation both quantities.

In Fig. 3 we show the blocking probability of a system with $m = 50$, $1/\mu = 30$ s and $P_r = 0.2$. These values have been selected as representative of reasonable operational conditions of a cellular network in which users wait on average 30 s before issuing a new request after a service is completed (e.g., the time needed to read a web page) and the operator allows maximum 50 users connected (this is a relatively small cell) and set the RRC timeout to ~ 6s (values used by the operators vary from a few seconds to a few tens of seconds).

In the figure we compare model and simulation, which are quite close. We also report the results computed with the approximate models presented in Section V-D, which behave quite poorly in all cases. The approximate models do not account for the presence of the infinite-server queue, that affects how and when customers return to service within the `RRC_CONNECTED` timeout. However, for large values of m, as we will comment later, larger fractions of the system population move to the *NETWORK* queue, and the presence of *THINK* becomes less important. Indeed, Fig. 4 shows that with $m = 200$ (which is for a large or dense cell, as today's base stations allow ~ 100 `RRC_CONNECTED` users) the approximate models can yield results comparable to our model, especially when using iterative methods. We remark that our model and simulation are very close under all parameter configurations, and that the lower-end of the blocking probability curves can be approximated only with our closed-form model or with iterative methods. Using the latter is less convenient in terms of complexity and because they cannot be used, e.g., to analytically tune the system.

Fig. 5 confirms that increasing m without changing μ and P_r has a beneficial impact on throughput, because more customers utilize the server of queue *NETWORK*. The figure also shows that the throughput cannot reach 100% of the *NETWORK* capacity c, unless a high number of parallel services are allowed.

Time spent at queue *NETWORK*. Fig. 6 displays the average time spent by customers in the system to complete service of their request, normalized to the average service time of a request when served with all *NETWORK* resources, i.e., normalized to

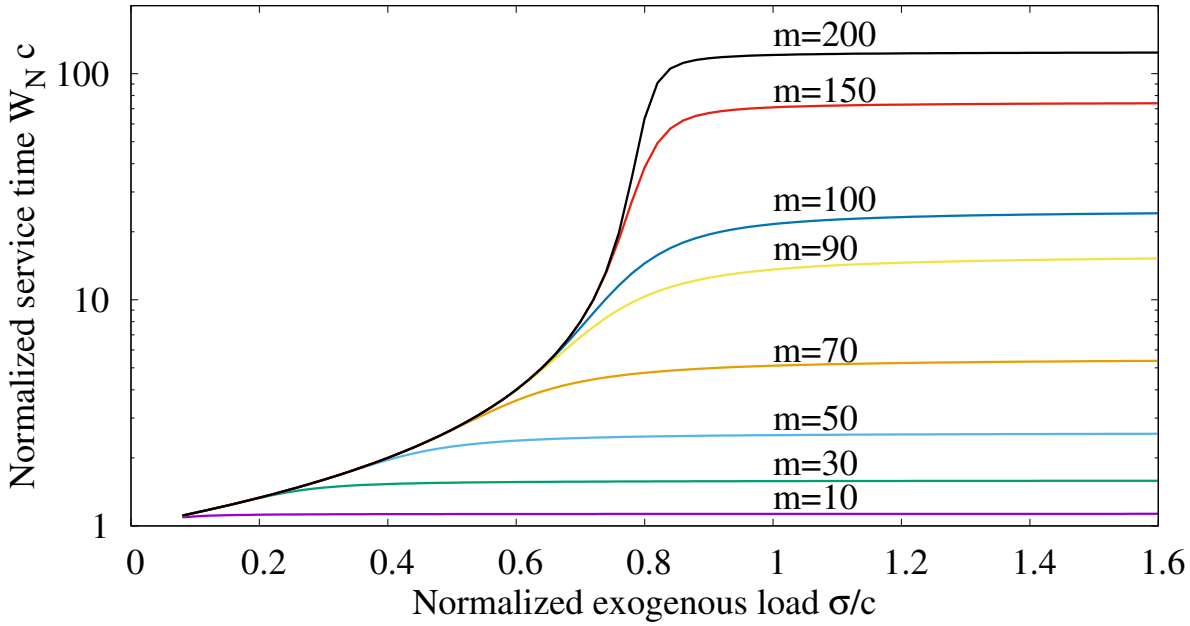

Fig. 6. The average time spent at queue *NETWORK* increases with the exogenous request arrival rate ($\mu^{-1} = 30$ s, $P_r = 0.2$)

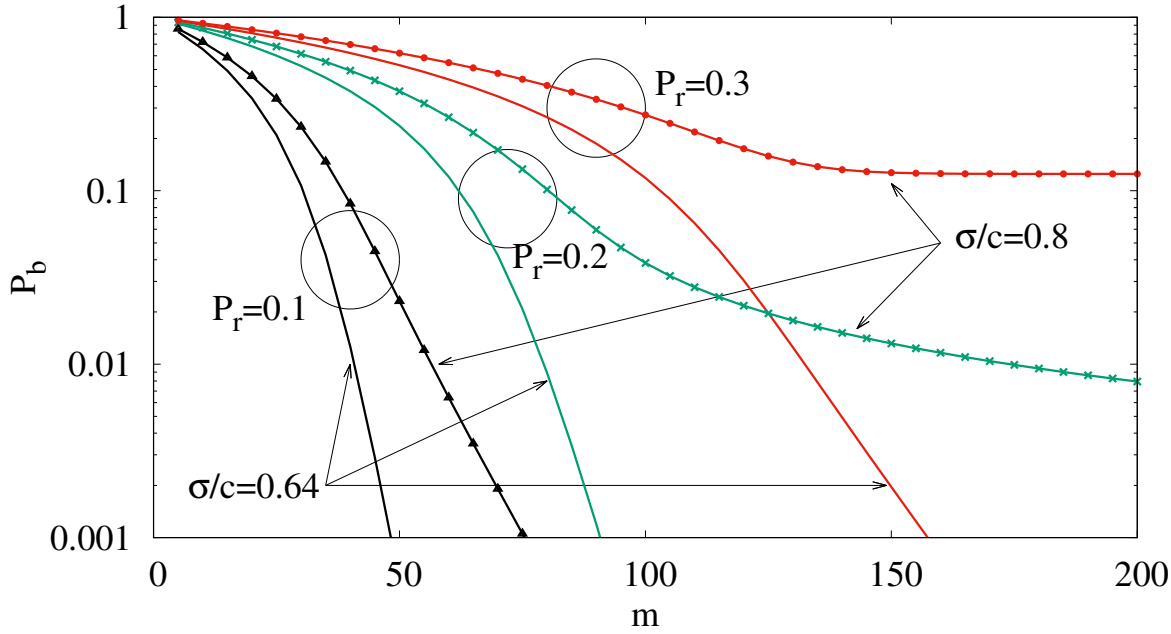

Fig. 7. Impact of m on blocking probability, with $\mu^{-1} = 30$ s

$1/c$. Since *NETWORK* uses a processor sharing discipline, the plotted values also represent the average number of simultaneously served requests. With $1/\mu = 30$ s and $P_r = 0.2$, the figure shows that the time necessary to complete a service saturates before the exogenous load reaches 100%, which is due to the presence of recently served customers with allocated resources. In the example, 20% of served customers return to service before the timeout expiration. Meanwhile, queue *NETWORK* keeps service positions ready for them, except they are unused, therefore imposing a restriction on the number of ongoing parallel services. For instance, with a load that saturates the throughput and m sufficiently high to reach the capacity c, using Little's law, the number of customers in *THINK* is $\sim c \cdot (P_r/\mu)$. With $m = 200$ and $1/c = 80$ ms (the time needed to serve a file of 1.5 MB with a 150 Mb/s downlink connection), the number of customers in `RRC_CONNECTED` status with no ongoing transmission is 75, i.e., 37.5% of m.

Impact of m. The importance of m is detailed in Fig. 7 for various values of the exogenous load and P_r, with $1/\mu = 30$ s. Increasing m always reduces the blocking probability, especially at high loads, although the price to pay for this improvement is a higher service time (see Fig. 6). That is, increasing m allows the system to keep more users busy, although for longer intervals, and reduces the blocking probability experienced by new freshly arrived customers.

Impact of P_r. The way customers returning before the RRC timeout affect blocking probability and throughput is detailed

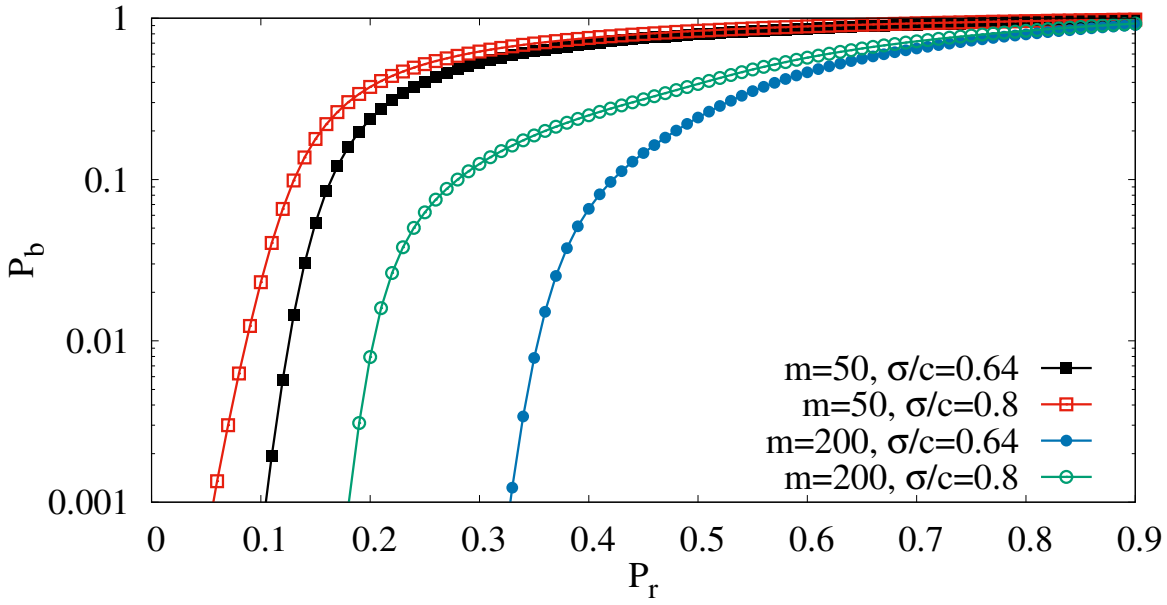

Fig. 8. Impact of returning customers on blocking probability, with $\mu^{-1} = 30$ s: high return rates worsen blocking

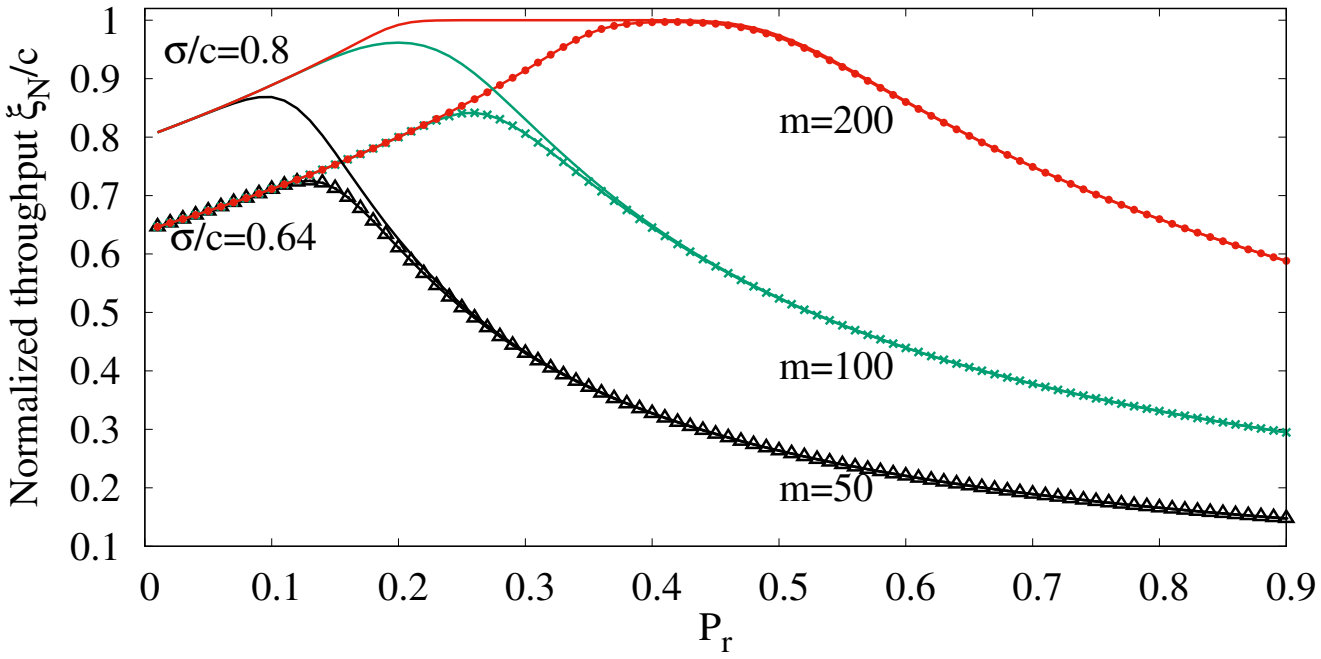

Fig. 9. Impact of returning customers on throughput: high return rates induce throughput drops ($\mu^{-1} = 30$ s)

in Fig. 8 and Fig. 9, respectively. Returning customers have a very negative impact on blocking probability, which is exactly the reason why we cannot analyze a cellular system without considering the RRC timeout and the returning customers, and the reason why operators need to finely tune the timeout values they use in their base stations. In terms of throughput, it is interesting to notice how reaching capacity c not only depends on m, but also on the fraction of returning customers. Indeed, a very high rate of returning customers heavily degrades the network performance, and the throughput collapses. The optimal value of P_r is hard to figure out, since it depends on other parameters such as m and the exogenous load. The figure shows that a network operator should adjust P_r (i.e., the RRC timeout) depending on cell parameters (m and c) and based on the traffic characteristics (σ and μ).

Relative importance of RRC timeout duration. To better understand the role of the RRC timeout T_{TO}, and considering that P_r is affected by the traffic characteristics through μ, we now plot the normalized network throughput as a function of both $1/\mu$ and T_{TO}.

Fig. 10 generalizes the analysis of the behavior of the curves of Fig. 9 for the case of intermediate offered loads ($\sigma/c = 0.64$) and $m = 200$. From the figure, we can observe that the base station capacity can be reached for either some optimal values of the timeout, or for very low values of $1/\mu$. In fact, low think times make improbable the return of customers without having to request a new connection through the random access procedure. As shown in Fig. 11, for higher loads ($\sigma/c = 0.8$), the impact

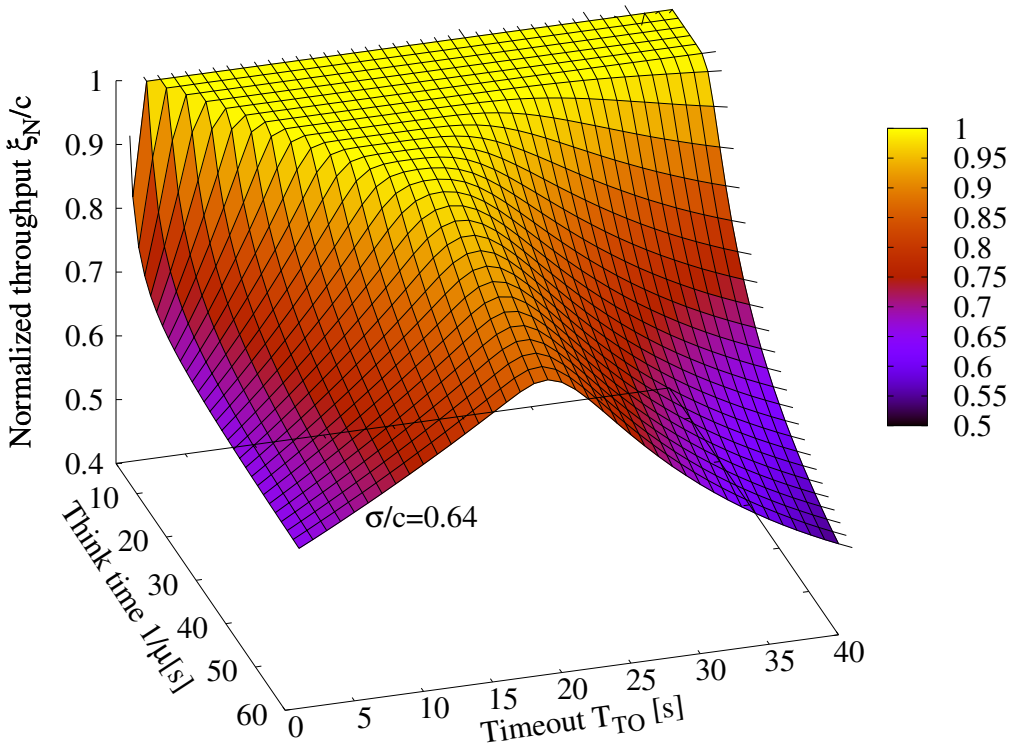

Fig. 10. Impact of timeout and think time, with $\sigma/c = 0.64$ and $m = 200$

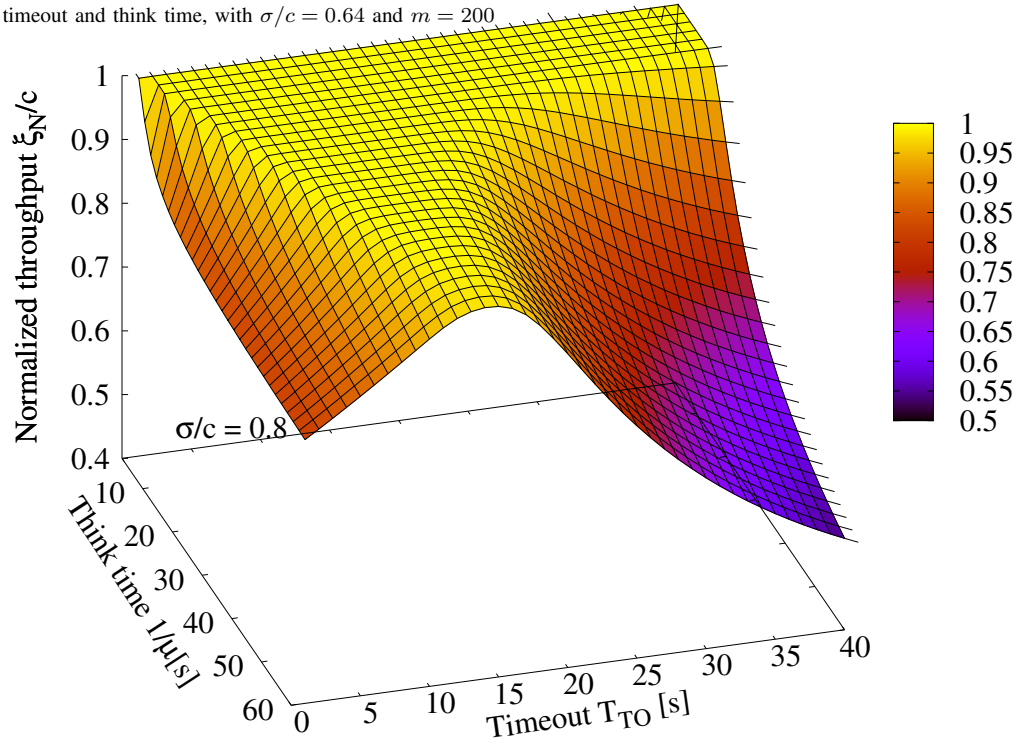

Fig. 11. Impact of timeout and think time, with $\sigma/c = 0.8$ and $m = 200$

of T_{TO} is very relevant, as it causes throughput drops at different values, depending on traffic conditions. However, both 3D plots show that for reasonably low average think times (e.g., below $15 - 20$ s), the choice of T_{TO} is much less critical.

These results can be used not only to design a network control mechanism to dynamically steer network configuration to follow user's demand, but also to support the design of machine-type communications in smart factories, in which the think time and the offered load can be imposed rather than estimated, and followed by the network.

VII. Related work

Outage probability has always been a key performance index in wired and wireless networks, starting with the early days of telephony. Its importance is becoming more and more evident, with the increased dependence of users on their smartphone

services. This is why availability is considered one of the key performance indicators for LTE and 5G [8], even if these new technologies provide much higher capacity and widespread coverage.

In recent works on the performance of cellular networks, such as [19], blocking probability becomes the focus of the analysis, relating outage probability to both cell size and data rates. In addition, the paper also analyses energy efficiency with respect to blocking probability instead of the more classical throughput.

In LTE networks, the concept of outage probability goes along with the reduction of latency: resources have to be made available to end users in the shortest possible time, even in extremely crowded environments. Guaranteeing high capacity, low latency and service availability are the challenges for next generations of cellular networks. In [6] a detailed evaluation based on real measurements of a popular public event is performed. Measurement results show that the number of dropped connections during such big events increases more than two orders of magnitude. Authors in [8] investigate more in detail the bottlenecks in the network, showing how performance drops in crowded scenarios are due to congestion of both Random Access resources and "network" resources. Again, in [6] and [8], outage probability is identified as one of the main roots of the problem.

Although the relevance of blocking probability has evolved together with cellular network standards, the way it can be computed in an easy and effective manner still remains a research challenge. The system behavior in the `RRC_CONNECTED` state has not been the subject of much research so far. The model we propose in this paper aims at tackling this problem in a simple and effective way.

VIII. Conclusions

The performance of cellular networks is driven by a set of elements whose effects are rather complex to predict. Characterizing and understanding the behavior and performance of these different elements is a prerequisite for any sound quantitative optimization and management of these networks.

This paper investigates the characteristics of one of the components of cellular networks which has been overlooked so far, i.e., the system behavior in the `RRC_CONNECTED` state. For this components we presented a simple, yet extremely accurate model, which can be solved in closed form, and for which it is possible to obtain recursive solutions in the maximum permitted number of simultaneous service instances.

The results of our model provide significant insight into the role played by system parameters such as the RRC timeout, and the persistence of the mobile users, showing how it is possible to achieve the best performance. As such, the model we presented in this paper can be an important tool to assist cellular network configuration and management, and to support network operators in network planning and design.

References

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021," Tech. Rep., February 2017.

[2] A. Q. Gill, D. Bunker, and P. Seltsikas, "Moving Forward: Emerging Themes in Financial Services Technologies Adoption," *Communications of the Association for Information Systems*, 2015.

[3] C. Torres-Huitzil and A. Alvarez-Landero, "Accelerometer-Based Human Activity Recognition in Smartphones for Healthcare Services," *Mobile Health, Springer Series in Bio-/Neuroimformatics*, vol. 5, pp. 147–169, 2015.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future generation computer systems*, vol. 29, no. 7, 2013.

[5] D.-H. Shin, "Measuring the Quality of Smartphones: Development of a Customer Satisfaction Index for Smart Services," *International Journal of Mobile Communications*, vol. 12, no. 4, pp. 311–327, 2014.

[6] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "Characterizing and Optimizing Cellular Network Performance During Crowded Events," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1308–1321, Jun. 2016.

[7] Y. J. Choi, S. Park, and S. Bahk, "Multichannel Random Access in OFDMA Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 603–613, Mar. 2006.

[8] P. Castagno, V. Mancuso, M. Sereno, and M. Ajmone Marsan, "Why Your Smartphone Doesn't Work in Very Crowded Environments," in *Proc. of WoWMoM*, 2017.

[9] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and The Road to 5G, Third Edition*, 3rd ed. Academic Press, 2016.

[10] C. Úbeda, S. Pedraza, M. Regueira, and J. Romero, "LTE FDD Physical Random Access Channel Dimensioning and Planning," in *Proc. of IEEE VTC Fall*, 2012.

[11] M. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE-A Random Access for Massive Machine-Type Communications in Smart Cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 76–83, January 2017.

[12] G. C. Madueño, J. J. Nielsen, D. M. Kim, N. K. Pratas, C. Stefanović, and P. Popovski, "Assessment of LTE Wireless Access for Monitoring of Energy Distribution in the Smart Grid," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 675–688, 2016.

[13] M. Ajmone Marsan, D. Roffinella, and A. Murru, "ALOHA and CSMA Protocols for Multichannel Broadcast Networks," in *Proc. of Canadian Commun. Energy Conf.*, Montreal, P.Q., Canada, Oct. 1982.

[14] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, and V. Casares-Giner, "Performance Analysis of Access Class Barring for Handling Massive M2M Traffic in LTE-A Networks," in *Proc. of IEEE ICC*, 2016.

[15] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *Journal of ACM*, vol. 22, no. 2, pp. 248–260, Apr. 1975.

[16] S. Asmussen and M. Bladt, "Renewal Theory and Queueing Algorithms for Matrix-Exponential Distributions," in *Matrix-analytic Methods in Stochastic Models,* Lecture Notes in Pure and Applied Mathematics, vol. 183, 1997, pp. 313–341.

[17] M. Reiser and S. S. Lavenberg, "Mean-Value Analysis of Closed Multichain Queuing Networks," *Journal of ACM*, vol. 27, no. 2, pp. 313–322, Apr. 1980.

[18] K. C. Sevcik and I. Mitrani, "The Distribution of Queuing Network States at Input and Output Instants," *J. ACM*, vol. 28, no. 2, 1981.

[19] S. Batabyal and S. S. Das, "Distance Dependent Call Blocking Probability, and Area Erlang Efficiency of Cellular Networks," in *Proc. of IEEE VTC Spring*, May 2012.